

ASYNCHRONOUS FAULT-TOLERANT
ENCLOSURE SERVICES INTERFACE

BACKGROUND OF THE INVENTION

The present invention is generally related to computer systems, and, more particularly, the present invention is related to bus interface techniques that enable asynchronous enclosure services between an enclosure processor and a host bus adapter.

5 Known bus and protocol interface techniques of most common interfaces for relatively small computer systems, such as ATA (Advanced Technology Attachment, also known in the art as Integrated Development Environment (IDE) interface), SATA (Serial Advanced Technology Attachment), UATA (Ultra ATA), etc., have no means for supporting an enclosure processor.

10 More particularly, disk drive enclosure service processors are known to use various standard protocols that unfortunately fail to address the needs of inexpensive interfaces such as ATA. For example, in SCSI Accessed Fault-Tolerant Enclosures (SAF-TE), the enclosure processor is just another device attached to the SCSI bus and may use a SAF-TE protocol that synchronously polls the enclosure processor at a
15 predefined rate, e.g., every two seconds. Performing synchronous polling in an ATA interface regardless of the condition of the devices in the enclosure could compromise the performance of the Host Bus Adapter (HBA) since such polling typically interrupts operation of the Central Processing Unit (CPU) with every character arrival or transmittal. Thus, interrupt overhead of a synchronous polling technique could
20 detrimentally affect the performance of the HBA in the ATA interface.

In SCSI Enclosure Services (SES), an elaborate command set, mainly used in FIBRE-based interfaces, requires the enclosure processor to be coupled to one or more of the Fibre drives. The command set relies on special diagnostics commands to be sent through one or more of the drives, which in turn relay such commands to the
25 enclosure processor. In an ATA interface it would be impractical to sacrifice one or more of the drives for providing communications with the enclosure processor. In fact, the ATA interface command protocol does not support any form of enclosure services.

In view of the foregoing discussion, it would be desirable to provide reliable and low-cost interface techniques between the enclosure processor and the HBA to carry out Enclosure Services in inexpensive and widely used interfaces for small computer systems.

5

BRIEF SUMMARY OF THE INVENTION

Generally, the present invention fulfills the foregoing needs by providing in one aspect thereof a method for enabling enclosure services in a computer system including a multi-device enclosure generally remote from a host bus adapter. The method provides a communications port between the multi-device enclosure and the host bus adapter. The method further provides a plurality of slots for removably receiving respective devices in the enclosure, with at least one of the devices comprising an Advanced Technology Attachment (ATA)-accessible device, and respective transceivers for asynchronously interconnecting the enclosure processor and the host bus adapter through the communications port. The method allows configuring the processor to asynchronously notify the host bus adaptor of the status of any given device of the enclosure upon the occurrence of predefined device events, with at least one of the events being selected from the group consisting of device insertion, device withdrawal, and malfunction indications regarding any of the devices of the multi-device enclosure.

The present invention further fulfills the foregoing needs by providing in another aspect thereof, a computer bus interface for enabling enclosure services in a computer system including a multi-device enclosure generally remote from a host bus adapter and including a plurality of slots for removably receiving respective devices in the enclosure. The interface includes a communications port between the multi-device enclosure and the host bus adapter. At least one of the devices of the multi-device enclosure comprises an Advanced Technology Attachment (ATA)-accessible device. The interface further includes a pair of transceivers for asynchronously interconnecting the enclosure processor and the host bus adapter through the communications port. Memory is provided for storing a plurality of instructions for configuring the processor to asynchronously notify the host bus adaptor of the status of any given device of the enclosure upon the occurrence of predefined device events,

with at least one of the events being selected from the group consisting of device insertion, device withdrawal, and malfunction indications regarding any of the devices of the multi-device enclosure.

In yet another aspect of the invention, a computer-readable medium including instructions for causing an interface to enable enclosure services in a computer system is provided. The multi-device enclosure includes a plurality of slots for removably receiving respective devices therein, with at least one of the devices comprising an Advanced Technology Attachment (ATA)-accessible device. The computer-readable medium comprises instructions for:

- configuring respective transceivers for asynchronously interconnecting the enclosure processor and a host bus adapter through a communications port; and
- configuring the processor to asynchronously notify the host bus adaptor of the status of any given device of the enclosure upon the occurrence of predefined operational events, with at least one of the events being selected from the group consisting of device insertion, device withdrawal, and malfunction indications regarding any of the devices of the multi-device enclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the invention when read with the accompanying drawings in which:

FIG. 1 illustrates a block diagram representation of a computer bus interface for enabling enclosure services in a computer system including a multi-device enclosure including a plurality of slots for removably receiving respective devices, such as one or more ATA-based devices.

FIGS. 2-4 illustrate respective flow charts of exemplary operational interrelationships regarding the computer bus interface of FIG. 1.

DETAILED DESCRIPTION OF THE INVENTION

GLOSSARY

| | | |
|----|----------|---|
| | FW | Firmware |
| | HBA | Host Bus Adapter, an I/O adapter that, for example, connects a host |
| 5 | | I/O bus to the storage memory system of the host |
| | HOT PLUG | The process of adding or removing a device from a bus while transactions involving other devices are occurring over the bus |
| | JBOD | Just a Bunch Of Disks – Used to refer to non-RAID components. |
| | RAID | Redundant Array of Independent Disks, a collection of two or more |
| 10 | | disks working together as a single independent disk drive |
| | RX | Receiver |
| | SCSI | Small Computer System Interface, a standard that defines connections between computers and peripheral devices |
| | SES | SCSI Enclosure Services, a standard for SCSI access to services within |
| 15 | | an enclosure including one or more SCSI devices, such as disk drives, power supplies, cooling elements, temperature sensors, etc. |
| | SPES | Serial Port Enclosure Services |
| | TX | Transmitter |
| | XOR | An exclusive OR (XOR) logic operation is true if only one of the |
| 20 | | inputs is true, but not both. |

OVERVIEW

EXEMPLARY HBA FUNCTIONAL ABILITIES

| | |
|----|--|
| | Query SPES Processor for version and configuration information |
| 25 | Set SPES Processor component states |
| | Query enclosure present component status |
| | Query enclosure present power supply status |
| | Query the present temperature of the enclosure |
| | Provide Active Cable Continuity checking (PING) |

EXEMPLARY SPES PROCESSOR FUNCTIONAL ABILITIES

Asynchronously notify HBA when any device has been inserted or removed from the back plane slots

Asynchronously notify HBA when one or more power supplies are out of spec or
5 have failed

Asynchronously notify HBA when one or more cooling fans are out of spec or have failed

Asynchronously notify the HBA when any enclosure temperature is out of range.

Optionally provide Active Cable Continuity checking (PING)

10 FIG. 1 illustrates a block diagram representation of a computer bus interface 10 for enabling enclosure services in a computer system including a multi-device enclosure 12 generally remote from a host bus adapter 14 and including a plurality of slots 16 for removably receiving respective devices, such as one or more ATA-based devices, represented by devices 1 through n, such as disk drives or any other computer
15 peripheral equipment. The enclosure 12 may further include additional devices that may vary depending on the specific requirements of any given application. Examples of such additional devices may include a fan 20, a power supply 22, a temperature sensor 24, lights, or audiovisual indicators 26 for communicating status information of the enclosure to a user.

20 A communications port 30 is provided between the multi-device enclosure and the host bus adapter. A pair of transceivers 32 and 34 is provided for asynchronously interconnecting an enclosure processor 40 and the host bus adapter 14 through the communications port. In one exemplary embodiment, communications port 30 comprises a serial communications port and each transceiver comprises a Universal
25 Asynchronous Receiver Transmitter (UART). A memory device 42 includes a plurality of instructions for configuring the processor 40 to asynchronously notify the host bus adaptor of the status of any given device of the enclosure upon the occurrence of predefined device events. Examples of the events may include device insertion, device withdrawal, and malfunction indications regarding any of the devices
30 of the multi-device enclosure. Memory 42 further includes instructions for configuring the host bus adapter to control, through the communications port, the enclosure processor to set respective device states of the multi-device enclosure. The

instructions in memory 42 also allow configuring the host bus adapter to generate a set of queries transmitted through the communications port and requiring response from the enclosure processor regarding the status of respective devices of the multi-device enclosure. As used herein the expression “enclosure services” refers to techniques for sensing and monitoring the physical and/or operational conditions of devices within an enclosure as well as allowing access to status reporting and configuration features of the enclosure, such as indicator LEDs of the enclosure, cooling equipment, insertion and removal of devices, etc. It will be understood that the term “enclosure” should not be construed narrowly since the concepts of the present invention are not dependent in any way on the existence of any remote physical enclosure. For example, it is contemplated that in some applications both the enclosure devices and the host bus adaptor could be integrated within a common chassis of a small computer, and thus the concepts of the present invention may be advantageously applied regardless of the existence of any separate physical enclosure.

EXEMPLARY HARDWARE REQUIREMENTS

As suggested above, in one exemplary embodiment, an RS-232C/422 serial interface is used for communication between the SPES processor and the HBA. It will be appreciated by those skilled in the art that other types of serial interfaces, e.g., Firewire (IEEE 1394) interface, parallel interface, etc., could have been used.

Character spacing of about 1ms per character is employed in one exemplary embodiment, this corresponds to a baud rate of about 9600 bps. Some of the factors for choosing this character spacing are as follows:

Presently, small 8-bit SPES processors are typically not very fast (about 800ns per instruction) and need time to perform the required XOR checksum functions of the protocol. This character spacing allows about 1 ms of processing time between each character.

HBA adapter performance could otherwise be impacted with serial interrupts and task switches.

EXEMPLARY PHYSICAL INTERCONNECT REQUIREMENTS

- In one exemplary embodiment the interconnect requirements are predefined to meet the applicable ATA interface requirements, such as ATA and Serial ATA type hot-plug enclosures. Assuming the user was to just randomly connect the disk cable to the HBA disk channels, the HBA firmware may not be able to map which enclosure slot is connected to which HBA channel.

For example, in a four slot ATA enclosure, the four disk drive cables should be connected to the HBA in the following manner:

- Cable for slot 0 would be connected to Channel 0 of the HBA
- 10 Cable for slot 1 would be connected to Channel 1 of the HBA
- Cable for slot 2 would be connected to Channel 2 of the HBA
- Cable for slot 3 would be connected to Channel 3 of the HBA

- The HBA and SPES processor assume this one-to-one interconnect mapping scheme. It will be appreciated by those skilled in the art, that such a mapping scheme may vary in accordance with the specific requirements of the interface being supported.

EXEMPLARY CONTROL PROTOCOL

ACTIVE CABLE CONTINUITY

- 20 Active Cable Continuity is used to verify the connection between the HBA and the SPES processor. In one exemplary embodiment, a PING character is sent periodically, e.g., once every 10 seconds, provided that a SPES transmission frame is not being transported. The sender sends the PING character and the recipient returns the invert (NOT PING) of that PING character. In this exemplary embodiment, the
- 25 PING character is defined as 0x2B and the NOT PING character is defined as its binary invert 0xD4.

The HBA is configured to support initiating the PING cycle with the SPES Processor and to return the NOT PING response. Optionally, the SPES Processor may also initiate the PING cycle with the HBA responding with the NOT PING response.

EXEMPLARY SPES TRANSMISSION FRAME

Table 1 below illustrates an exemplary encapsulation of SPES command and data payloads using a straightforward data format using 8-bit ASCII characters.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|---------------------------------|---|---|---|---|---|---|---|
| Byte | | | | | | | | |
| 0 | Start Of Header-SOH (0x01) | | | | | | | |
| 1 | Total TX byte count | | | | | | | |
| 2~n-3 | SPES command and page payload | | | | | | | |
| n-2 | XOR checksum LO | | | | | | | |
| n-1 | XOR checksum HI | | | | | | | |
| n | End Of Transmission -EOT (0x04) | | | | | | | |

TABLE 1

START OF HEADER (SOH)

- 5 ASCII character (0x01) marks the beginning of each serial transfer packet. This is the first byte sent to the recipient.

TOTAL TX BYTE COUNT

- This count is the number of bytes to be received starting from byte 2 through the EOT byte. In one exemplary embodiment, this is the first byte to be part of the XOR checksum.

SPES COMMAND AND PAGE DATA

The SPES command and page data may be built by either the HBA or the SPES Processor. In one exemplary embodiment, this is the last byte of the page payload data and is the last byte of XOR checksum protected data.

- 15 XOR CHECKSUM HI AND LO

- To ensure adequate protection of the page payloads, it was felt that a 16-bit CRC could appropriately protect the data. However, such an implementation often requires multiplication instructions or a lookup table and, thus, could become somewhat burdensome, or expensive, for 8-bit microprocessors typically used in enclosures. Thus, in one exemplary embodiment, the serial data stream is protected using an uncomplicated but effective scheme that can be implemented by even small

8-bit microprocessors. The proposed scheme is believed to more than adequately detect any errors in the transmitted SPES transmission frame.

As suggested above, the first byte of data undergoing a logical XOR operation is the TX byte count and the last byte of the page payload. In one exemplary embodiment, a 16-bit XOR checksum is “produced” by the TX driver in the following manner:

A 16-bit result value is initialized to zero when the first protected byte is about to be transmitted. As each byte is transmitted, the 16-bit XOR checksum value is rotated one bit position to the left. When the rotation is complete, the previous high order bit [15] would appear at bit position [0]. The transmitted byte then undergoes a XOR operation with the low order byte [7:0] of the 16-bit result value. This continues until all protected bytes have been transmitted. The produced two-byte XOR checksum result is inverted and then transmitted. First the lower byte [7:0] and then the high order byte [15:8].

Similarly, a 16-bit XOR checksum is “verified” by the RX driver in the following manner:

When the first byte of protected data arrives, the upper byte of the XOR checksum is set to zero and the lower order byte is set equal to the first byte of protected data. As each additional byte arrives, the 16-bit XOR checksum value is rotated one bit position to the left. When the rotation is complete, the previous high order bit [15] shall now appear at bit position [0]. The newly arrived byte then undergoes a XOR operation with the low order byte [7:0] of the XOR checksum.

When both bytes of the senders transmitted XOR value are received, these values undergo a XOR operation with the receiver calculated values and should produce a final value of 65,535 if the transmission is valid.

END OF TRANSMISSION (EOT)

The ASCII 0x04 character marks the end of SPES transmission frame. Once the EOT has been transmitted, the recipient is given an appropriate period of time (e.g., two seconds) to return an ASCII ACK (0x06) or NAK (0x15) character response to the sender.

EXEMPLARY SPES COMMAND FRAME

Table 2 below lists exemplary SPES page payloads encapsulated with a Status and Command byte forming a SPES command frame. All commands generally require a SPES page payload even if it is only the page identifier.

5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-------------------|---|---|---|---|---|---|---|
| Byte | | | | | | | | |
| 0 | SPES Status | | | | | | | |
| 1 | SPES Command | | | | | | | |
| 2~n | SPES page payload | | | | | | | |

TABLE 2

SPES STATUS

This is the completion status of any SPES command. Typically, the recipient fills in this status but in cases of transmission failures the TX driver of the sender will fill it in with a TX_type error status.

SPES_COMPLETED (0x00)

This indicates to the sender that the recipient received a valid command and it was successfully completed. The Page identifier within the payload indicates which page was successfully updated. Actual page data may or not be attached depending on the original command sent to the recipient.

SPES_ERROR (0x01)

SPES command completed with unknown error. For ease of debugging, the recipient should return the entire page data payload back to the sender.

SPES_PARM_ERROR (0x02)

The attached page payload has one or more bytes that are incorrect. The page identifier within the payload is valid. For ease of debugging, the recipient should return the entire page data payload back to the sender.

SPES_NOT_SUPPORTED (0x03)

This indicates to the sender that the SPES command or page is not supported. For ease of debugging, the recipient should return the entire page data payload back to the sender.

5 SPES_BUSY (0x04)

The recipient received the valid SPES command but was unable to process at this time. The sender needs to retry again later. The sender should wait a suitable period of time (e.g., five seconds) before retrying the SPES command that caused this SPES_BUSY status to be returned.

10 SPES_TX_TIMEOUT(0x80)

The SPES serial transmission frame was sent but the required serial ACK/NAK response was not received from the recipient within a suitable time period, e.g., about two seconds, after sending the EOT character of the SPES frame. In this case the senders' own TX driver returned the entire SPES command and payload to the senders completion queue.

15 SPES_TX_NAK(0x81)

This error status indicates that the recipient received the SPES transmission frame and responded with a NAK response. This error may mean the XOR checksum was incorrect. In this error case the senders' own TX driver returned the entire SPES command and payload to the senders completion queue. Optionally, the TX driver is allowed to do one retry before returning this error.

20 EXEMPLARY SPES PAGE COMMANDS

The command byte instructs the recipient how to handle the request.
HBA REQUEST FOR SPES PAGE DATA (0X00)

25 The HBA is requesting the SPES Processor to return specific SPES page data. Only the page identifier is part of the payload. The SPES Processor should respond within an appropriate waiting period, e.g., about five seconds, by returning the requested page and status using SPES command "SPES Acknowledgement (0xFF)".
HBA UPDATE OF SPES PAGE DATA (0X01)

30 The HBA is requesting the SPES Processor to update the provided SPES page. Page identifier and data is part of the payload. The recipient should update internal page data and act upon on new page data. The SPES Processor should return SPES

Status using SPES command “SPES Acknowledgement (0xFF)” within an appropriate waiting period, e.g., about five seconds.

SPES PROCESSOR NOTIFY (0X82)

- 5 The SPES Processor is notifying the HBA that the attached page has changed. The HBA should return the SPES Status using SPES command “HBA ACKNOWLEDGEMENT (0X7F)” within an appropriate waiting period, e.g., about five seconds.

SPES Acknowledgement (0xFF)

- 10 The SPES Processor sends completion status of a previous request. The Page Code Identifier indicates the page being acknowledged and SPES status is the completion status. It is not necessary to return the page data as part of the payload unless the original SPES command required its return.

HBA ACKNOWLEDGEMENT (0X7F)

- 15 The HBA sends completion status of a previous SPES Processor request. The Page Code Identifier and the SPES command status must be returned. It is not necessary to return the page data as part of the payload unless the original SPES command required its return.

EXEMPLARY SPES PAGES

- 20 This section defines each of the SPES pages supported by this interface. Each page has a unique Page Code Identifier.

READ ENCLOSURE CONFIGURATION PAGE (0X00)

- As shown in Table 3 below, this page provides the HBA a means to query the SPES Processor features and manufacture information. The HBA will not allow unsolicited returns of this page by the SPES Processor. The SPES Processor will receive a negative acknowledgment frame and shall not consider this response from the HBA as a fatal condition.
- 25

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|--|-------------------------------|---|---|------------------------------|---|---|---|
| Byte | | | | | | | | |
| 0 | Page Code Identifier (0x00) | | | | | | | |
| 1 | SPES spec major version | | | | SPES spec minor version | | | |
| 2 | SPES device FW major version | | | | SPES device FW minor version | | | |
| 3~10 | 8 byte SPES Device Manufacture Name in ASCII | | | | | | | |
| 11~26 | 16 byte Model Name in ASCII | | | | | | | |
| 27~42 | 16 byte SPES device serial number | | | | | | | |
| 43 | Number of device slots | | | | | | | |
| 44 | Celsius | Number of temperature sensors | | | | | | |
| 45 | Number of Fans | | | | | | | |
| 46 | Number of Power supplies | | | | | | | |

TABLE 3

Byte 0: This byte is generally set to zero.

Byte 1: This byte indicates to what level of the SPES spec the SPES device will adhere to. In one exemplary embodiment, the byte is divided into two 4-bit fields.

For example, the upper 4 bits may represent the MAJOR version number and the lower 4 bits may represent the MINOR version. Valid range of each 4-bit field may be 0x0-0xF.

Example 1: If the version was version 1.4 then bits [7:4] would be set to 0x1 and bits [3:0] would be set to 0x4 which results in the byte being equal to 0x14.

Example 2: If the version was version 10.6 then the bits [7:4] would be set to 0xA and bits [3:0] would be set to 0x6 which results in the byte being equal to 0xA6.

Byte 2: This indicates the FW version level of the SPES device. The byte is divided into two 4-bit fields. As suggested above, the upper 4 bits may be the MAJOR

version number and the lower 4 bits may be the MINOR version. Valid range of each 4-bit field shall be 0x0-0xF. This byte has the same format as byte 1 above.

Byte 3~10: Eight bytes of the SPES processor Manufacturer's name. The name may be in a printable ASCII format. Unused bytes may be set to an ASCII space (0x20).

Byte 11~26: Sixteen bytes of the SPES processor's Model name. The name may be in a printable ASCII format. Unused bytes may be set to an ASCII space (0x20).

Byte 27~42: Sixteen bytes of the SPES processor's Serial Number. The serial number string may be in a printable ASCII format. Unused bytes may be set to an ASCII

5 space (0x20).

Byte 43: The number of device slots in the enclosure. In one exemplary embodiment, valid range shall be from 1-255. All enclosures should have at least one device slot.

10 Byte 44: Bits [6:0] indicates the number of temperature sensors that are reportable to the HBA. By way of example, valid range may be 0-127. The Celsius flag shall be set to indicate that the temperature fields returned by the processor are in Celsius. If this flag is cleared, then the temperatures returned will be in Fahrenheit. A count of zero shall mean the enclosure has no reportable temperature sensors.

Byte 45: The number of possible reportable fans within the enclosure. By way of example, valid range may be 0-255. Fans not installed and reportable shall be
15 included in this count. A count of zero shall mean the enclosure has no reportable fans.

Byte 46: The number of possible reportable power supplies within the enclosure. By way of example, valid range may be 0-255. Power supplies no installed and reportable must be included in this count. A count of zero shall mean the enclosure
20 has no reportable power supplies.

EXEMPLARY ENCLOSURE STATUS PAGES

Operational status of the enclosure components may be read by the HBA or delivered by the SPES Processor using the following Enclosure Status pages. The
25 SPES Processor generally will not be allowed to deliver any of these pages until the HBA has performed the "HBA REQUEST FOR SPES PAGE DATA (0X00)" function and has set aside enough memory for incoming status pages. The number of status bytes available for transfer to the HBA will depend on the number of reportable components reported in the "HBA REQUEST FOR SPES PAGE DATA (0X00)".

30 Table 4 below illustrates an exemplary Fan status page (0x01)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----------------------------|---|---|---|---|---|---|---|
| Byte | | | | | | | | |
| 0 | Page Code Identifier (0x01) | | | | | | | |
| 1 | Number of status entries | | | | | | | |
| 2~n | Fan Status | | | | | | | |

TABLE 4

Byte 0: Generally set to 0x01

Byte 1: Number of status entries to follow. A count of zero shall mean that the

5 enclosure has no reportable fans.

Byte 2~n the Status of each fan:

Exemplary Valid fan status values:

0x00 Fan is operational

10 0x01 Fan is malfunctioning

0x02 Fan is not installed

0x80 Unknown Status or Status Not Reportable

Table 5 below illustrates an exemplary Power Supply status page (0x02)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----------------------------|---|---|---|---|---|---|---|
| Byte | | | | | | | | |
| 0 | Page Code Identifier (0x02) | | | | | | | |
| 1 | Number of status entries | | | | | | | |
| 2~n | Power supply Status | | | | | | | |

TABLE 5

Byte 0: Generally set to 0x02

Byte 1: Number of status entries to follow. A count of zero shall mean that the enclosure has no reportable power supplies.

20 Byte 2~n the Status of each power supply:

Valid power supply status values:

0x00 Power supply is operational

0x01 Power supply is malfunctioning

0x02 Power supply is not installed

- 5 0x80 Unknown Status or Status Not Reportable

Table 6 below illustrates an exemplary Device Slot Status (0x03)

The SPES Processor sends this frame to the HBA when the device slot status changes. The HBA does not have to poll this page but can optionally do so at anytime.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----------------------------|----|----|----|-----|-----|----|----|
| Byte | | | | | | | | |
| 0 | Page Code Identifier (0x03) | | | | | | | |
| 1 | Slot number | | | | | | | |
| 2 | NC | PF | FF | PC | IDA | IFA | DF | NE |
| 3 | | IF | HS | MA | EA | RS | DR | BF |
| 4 | | | | | | | DO | DP |

TABLE 6

Byte 0: Generally set to 0x03

Byte 1: Device slot number that is being reported by the SPES Processor or requested by the HBA. The number may be zero-based. That means the first slot within the enclosure may be selected to have a value of 0x00. By way of example, valid range may be 0-255.

Byte 2: This byte is a copy of the last one sent to the SPES processor by the HBA using the page. In one exemplary embodiment, the fields are defined as follows:
NE – No Error Flag. SET if there are no outstanding error conditions on this device slot.

DF – Device Faulty Flag. SET if this device has exhibited some hardware or data fault. This device is offline and is considered not functional.

IFA – In Failed Array Flag. SET if this device is in an array that is not recoverable and is classified as a dead array member.

IDA – In Degraded Array Flag. SET if this device is in an array that was previously fault-tolerant and has become non-fault-tolerant. Non-redundant configurations such as RAID 0 are inherently non-fault-tolerant, and should not make use of this flag.

PC – Parity Check Flag. SET if the device is in an array, which is undergoing parity check operation.

FF – Formatting Flag. SET for a slot that is currently in a low level formatting state.

PF – Predicted Fault Flag. SET if a fault prediction algorithm has determined that this device is likely to fail in the near future.

NC – Not Configured Flag. SET if a device is currently not configured as an element of an array or as a hot spare. (JBOD)

Byte 3: This byte is a copy of the last one sent to the SPES Processor by the HBA using the page. In one exemplary embodiment, the fields are defined as follows:

BF – Building Flag. SET for a slot that is currently in a building state. All members within the building array shall have this bit set. Non-redundant configurations, such as RAID 0 are inherently non-fault-tolerant, and should not make use of this flag.

DR – Device Rebuilding Flag. SET if the device is being rebuilt. Other drives in this array remain In Degraded Array State until the rebuild operation is complete. Non-redundant configurations such as RAID 0 are inherently non-fault-tolerant, and should not make use of this flag.

RS – Rebuild Stopped Flag. SET for a slot containing a device which was rebuilding, but the rebuild terminated abnormally or unsuccessfully. This flag and the Device Rebuilding (DR) flag cannot both be set for the same slot.

EA – Expanding Array Flag. SET for a slot that is currently expanding. All members in an expanding array shall have the bit set.

MA – Morphing Array Flag. SET for a slot that is currently morphing from one array type to another. All members of a morphing array shall have this bit set.

HS – Hot Spare Flag. SET if device in this slot is configured as a hot spare.

IF – Identify Flag. When SET, the SPES Processor is to flash the slots identifier signal for a suitable period of time (e.g., not less than 10 seconds and not more than 60 seconds, the identify period). This helps the user locate the device slot. If the device slot signal is being used for other purposes, then it will be temporally

suspended only for the identify flash period. After the identify period is over, this bit is auto-cleared by the SPES Processor. The HBA can clear this bit early with a subsequent command.

- 5 Byte 4: This byte contains flags that indicate conditions of interest on the slot, which the SPES Processor shall report to the HBA.

DP- Device Present Flag – This flag indicates whether or not there is a physical device inserted in the device slot. This does not imply that the device is ready for access on the bus. If the bit is CLEAR, no device is inserted at this time. If the bit is SET, a device is inserted.

DO- Device Operation Flag – This flag indicates whether or not the slot is activated so that the inserted drive may be accessed on the bus. SET means “ready” and CLEAR means “not ready.”

- 15 **Table 7 below illustrates an exemplary page for Write Device Slot Status (0x83)**
The Write Device Slot Status page is used by the HBA to inform the SPES Processor of the state of each slot and the device potentially inserted. This information is used to drive the enclosure visual and audible devices (LEDs, LCD, audible alarm, etc.) to some meaningful state, depending on the vendor’s implementation of such
- 20 LEDs/LCD and Alarms sub-systems.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----------------------------|----|----|----|-----|-----|----|----|
| Byte | | | | | | | | |
| 0 | Page Code Identifier (0x83) | | | | | | | |
| 1 | Slot number | | | | | | | |
| 2 | NC | PF | FF | PC | IDA | IFA | DF | NE |
| 3 | | IF | HS | MA | EA | RS | DR | BF |

TABLE 7

Byte 0: Generally set to 0x83

Byte 1: Device slot number that is being updated by the HBA. The number may be zero-based. That means the first slot within the enclosure may have a value of 0x00. For example, valid range may be 0-255.

- 5 Byte 2: These flags are used to inform the SPES Processor as to the state of the device slot. The SPES Processor shall update visual and alarm indicators to reflect the new state of the slot. This byte shall be retained by the SPES Processor and returned as part of the above as to the definition of bits within this byte.

- 10 Byte 3: These flags are used to inform the SPES Processor as to the state of the device slot. The SPES Processor shall update visual and alarm indicators to reflect the new state of the slot. This byte shall be retained by the SPES Processor and returned as part of the page above as to the definition of bits within this byte.

Table 8 below illustrates an exemplary Temperature status page (0x04)

15

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-------------------------------|---|---|---|---|---|---|---|
| Byte | | | | | | | | |
| 0 | Page Code Identifier (0x04) | | | | | | | |
| 1 | Number of temperature entries | | | | | | | |
| 2~n | Present temperature reading | | | | | | | |

TABLE 8

Byte 0: Generally set to 0x04

- 20 Byte 1: Number of entries to follow. A count of zero shall mean that the enclosure has no reportable temperature sensors.

- 25 Bytes 2~n: Each of these fields contains an unsigned integer indicating internal or external enclosure temperature at this sensor in degrees. This value, from 0 to 255, is based on a scale starting at -10 degrees. This gives an effective reportable temperature range from -10 degrees to 245 degrees. The returned value can be defined as Celsius or Fahrenheit.

For readers desirous of graphical representations, FIGS. 2-4 represent respective flow charts illustrative of exemplary operational interrelationships regarding the computer bus interface that, in accordance with aspects of the present invention, enables enclosure services in a computer system including a multi-device enclosure. More particularly, the flow chart depicted in FIG. 2 illustrates an exemplary interplay between the host bus adapter and the enclosure processor. FIG. 3 is a flow chart illustrative of receiver operation of either the host bus adapter or the enclosure processor, and FIG. 4 is a flow chart illustrative of transmitter operation of either the host bus adapter or the enclosure processor. The concepts illustrated in the flow charts of FIGS. 2 and 4 are based on the discussion provided in the context of Tables 1-8 above and for the sake of avoiding tiresome repetition to those skilled in the art such discussion will not be repeated.

The present invention can be embodied in the form of computer-implemented processes and apparatus for practicing those processes. The present invention can also be embodied in the form of computer program code containing computer-readable instructions embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other computer-readable storage medium, wherein, when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. The present invention can also be embodied in the form of computer program code, for example, whether stored in a storage medium, loaded into and/or executed by a computer, or transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via electromagnetic radiation, wherein, when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. When implemented on a general-purpose computer, the computer program code segments configure the computer to create specific logic circuits or processing modules.

While the preferred embodiments of the present invention have been shown and described herein, it will be obvious that such embodiments are provided by way of example only. Numerous variations, changes and substitutions will occur to those of skill in the art without departing from the invention herein. Accordingly, it is

intended that the invention be limited only by the spirit and scope of the appended claims.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
22